AMENDMENTS TO THE CLAIMS

Please amend claims 1, 13, 25 and 37 as follows.

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (currently amended) Apparatus for processing data comprising:

processing logic operable to perform data processing operations; and

an instruction decoder ~~operable to decode~~for decoding program instructions to control said processing logic to perform data processing operations specified by said program [[instructions;]] instructions, wherein said instruction decoder, ~~is responsive~~ in response to a compare and branch instruction, comprises a decoder for:

(i) ~~to perform~~performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) for copying, in dependence upon a result of said comparison, a program counter value to a third register;

[[(ii) to determine]] (iii) for determining a target branch address from a pre-programmed stored value and said program counter value; and

[[(iii) to branch]] (iv) for branching to a sub-routine at said target branch address in dependence upon a result of said comparison.

2. (original) Apparatus as claimed in claim 1, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

3. (original) Apparatus as claimed in claim 1, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

4. (original) Apparatus as claimed in claim 2, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

5. (original) Apparatus as claimed in claim 4, wherein said comparison determines whether said reference value is greater than or equal to said test value.

6. (original) Apparatus as claimed in claim 4, wherein said result of said comparison is determined from a carry flag value and zero flag value.

7. (original) Apparatus as claimed in claim 2, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

8. (original) Apparatus as claimed in claim 1, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

9. (original) Apparatus as claimed in claim 1, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

10. (original) Apparatus as claimed in claim 1, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

11. (original) Apparatus as claimed in claim 10, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

12. (original) Apparatus as claimed in claim 1, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

13. (currently amended) A method of processing data with an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode ~~program instructions~~a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said method comprising the steps of:

~~in response to a compare and branch instruction decoded by said instruction decoder controlling said processing logic:~~

(i) ~~to perform~~performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) copying a program counter value, in dependence upon said comparison, to a third register;

[[(ii)]] (iii) to determine a target branch address from a pre-programmed stored value and said program counter value; and

[[(iii)]] (iv) to branch to a sub-routine at said target branch address in dependence upon a result of said comparison.

14. (original) A method as claimed in claim 13, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

15. (original) A method as claimed in claim 13, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

16. (original) A method as claimed in claim 14, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

17. (original) A method as claimed in claim 16, wherein said comparison determines whether said reference value is greater than or equal to said test value.

18. (original) A method as claimed in claim 16, wherein said result of said comparison is determined from a carry flag value and a zero flag value.

19. (original) A method as claimed in claim 14, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

20. (original) A method as claimed in claim 13, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

21. (original) A method as claimed in claim 13, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

22. (original) A method as claimed in claim 13, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

23. (original) A method as claimed in claim 22, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

24. (original) A method as claimed in claim 13, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

25. (currently amended) A computer program product comprising a computer-readable storage medium including a computer program operable to control an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode ~~program instructions~~a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said computer program comprising the steps of:

~~a compare and branch instruction decodable by said instruction decoder to control said processing logic:~~

(i) ~~to perform~~performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) copying a program counter value, in dependence upon said comparison, to a third register;

[[(ii) to determine]] (iii) determining a target branch address from a pre-programmed stored value; and

[[(iii) to branch]] (iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison.

26. (original) computer program product as claimed in claim 25, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

27. (original) A computer program product as claimed in claim 25, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

28. (original) A computer program product as claimed in claim 26, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

29. (original) A computer program product as claimed in claim 28, wherein said comparison determines whether said reference value is greater than or equal to said test value.

30. (original) A computer program product as claimed in claim 28, wherein said result of said comparison is determined from a carry flag value and a zero flag value.

31. (original) A computer program product as claimed in claim 26, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

32. (original) A computer program product as claimed in claim 25, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

33. (original) A computer program product as claimed in claim 25, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

34. (original) A computer program product as claimed in claim 25, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

35. (original) A computer program product as claimed in claim 34, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

36. (original) A computer program product as claimed in claim 25, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.

37. (currently amended) A computer program product comprising a computer-readable storage medium including a computer program operable to translate non-native program instructions to form native program instructions directly decodable by an apparatus for processing data having processing logic operable to perform data processing operations and an instruction decoder operable to decode ~~program instructions~~a compare and branch instruction to control said processing logic to perform data processing operations specified by said program instructions, said native program instructions comprising:

~~a compare and branch instruction decodable by said instruction decoder to control said processing logic:~~

(i) ~~to perform~~ performing a comparison between a first value stored in a first register and a second value stored in a second register;

(ii) copying a program counter value, in dependence upon said comparison, to a third register;

[[(ii) to determine]] (iii) determining a target branch address from a pre-programmed stored value; and

[[(iii) to branch]] (iv) branching to a sub-routine at said target branch address in dependence upon a result of said comparison.

38. (original) A computer program product as claimed in claim 37, wherein said instruction is an array bounds checking instruction and said sub-routine is an array bounds exception handling routine.

39. (original) A computer program product as claimed in claim 37, wherein at least one of said first register and said second register are specified within said compare and branch instruction.

40. (original) A computer program product as claimed in claim 38, wherein said first value is a reference value specifying an array size and said second value is a test value determined from a decoded program instruction.

41. (original) A computer program product as claimed in claim 40, wherein said comparison determines whether said reference value is greater than or equal to said test value.

42. (original) A computer program product as claimed in claim 40, wherein said result of said comparison is determined from a carry flag value and a zero flag value.

43. (original) A computer program product as claimed in claim 38, wherein said branching operation comprises copying a pointer to said array bounds exception handling routine into a register specifying a next program instruction.

44. (original) A computer program product as claimed in claim 37, wherein said data processing apparatus comprises a co-processor and said pre-programmed stored value is read from a register of said co-processor.

45. (original) A computer program product as claimed in claim 37, wherein said compare and branch instruction is executed within a single processing cycle of said data processing apparatus when the branch is not taken.

46. (original) A computer program product as claimed in claim 37, wherein said instruction decoder is operable to decode translated platform-independent program instructions.

47. (original) A computer program product as claimed in claim 46, wherein said platform independent program instructions are one of :

Java bytecodes;

.net bytecodes;

MSIL bytecodes; and

CIL bytecodes.

48. (original) A computer program product as claimed in claim 37, wherein said data processing apparatus is operable in a user mode and a privileged mode and said data processing apparatus remains in said user mode during execution of said compare and branch instruction.